using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using PizzaModelsLibrary;

using PizzaDALLibrary;

namespace PizzaFEConsoleApp

{

class ManageEmployee

{

List<Employee> employees;

EmployeeDAL employeeDAL;

public ManageEmployee()

{

employeeDAL = new EmployeeDAL();

}

public void AddEmployees()

{

Employee employee = new Employee();

employee.GetEmployeeDetails();

try

{

employeeDAL.InsertNewEmployee(employee);

}

catch (Exception e)

{

Console.WriteLine("Could not add the employee");

Console.WriteLine(e.Message);

}

}

public void EditEmployeeName()

{

int id = GetEmployeeIdFromUser();

Employee employee = GetEmployeeById(id);

if(employee == null)

{

Console.WriteLine("Invalid Id. Cannot edit");

return;

}

Console.WriteLine("The employee you have chosen to edit name : ");

PrintEmployee(employee);

Console.WriteLine("Please enter the name : ");

employee.Name = Console.ReadLine();

if(employeeDAL.UpdateEmployeeName(id, employee.Name))

Console.WriteLine("Updated. New Details");

PrintEmployee(employee);

}

private void PrintEmployee(Employee employee)

{

Console.WriteLine("--------------------------");

Console.WriteLine(employee);

Console.WriteLine("--------------------------");

}

private Employee GetEmployeeById(int id)

{

List<Employee> employees = GetAllEmployee();

Employee employee = employees.SingleOrDefault(ep => ep.EmployeeId == id);

return employee;

}

private List<Employee> GetAllEmployee()

{

employees = null;

try

{

employees = employeeDAL.GetAllEmployees().ToList();

return employees;

}

catch (Exception)

{

throw;

}

}

private int GetEmployeeIdFromUser()

{

Console.WriteLine("Please enter the employee id");

int id;

while (!int.TryParse(Console.ReadLine(), out id))

{

Console.WriteLine("Invalid entry ID. Please try again...");

}

return id;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Data;

using PizzaModelsLibrary;

namespace PizzaDALLibrary

{

public class EmployeeDAL

{

private SqlConnection conn;

public EmployeeDAL()

{

conn = MyConnection.GetConnection();

}

public bool InsertNewEmployee(Employee employee)

{

SqlCommand cmd = new SqlCommand("proc\_InsertEmployee", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@ename", employee.Name);

cmd.Parameters.AddWithValue("@eage", employee.Age);

cmd.Parameters.AddWithValue("@deleted", 0);

conn.Open();

if (cmd.ExecuteNonQuery() > 0)

return true;

return false;

}

public ICollection<Employee> GetAllEmployees()

{

List<Employee> employees = new List<Employee>();

DataSet ds = new DataSet();

SqlDataAdapter adapter = new SqlDataAdapter("proc\_GetAllEmployee", conn);

adapter.SelectCommand.CommandType = CommandType.StoredProcedure;

adapter.Fill(ds);

Employee employee;

if (ds.Tables[0].Rows.Count == 0)

throw new NoWorkerException();

foreach(DataRow item in ds.Tables[0].Rows)

{

employee = new Employee();

employee.EmployeeId = Convert.ToInt32(item[0]);

employee.Name = item[1].ToString();

employee.Age = Convert.ToInt32(item[2]);

employee.IsDeleted = Convert.ToByte(item[3]);

employees.Add(employee);

}

return employees;

}

public bool UpdateEmployeeName(int id, string name)

{

SqlCommand cmd = new SqlCommand("proc\_UpdateEmployeeName", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@id", id);

cmd.Parameters.AddWithValue("@name", name);

conn.Open();

if (cmd.ExecuteNonQuery() > 0)

return true;

return false;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace PizzaModelsLibrary

{

public class Employee

{

public int EmployeeId { get; set; }

public string Name { get; set; }

public int Age { get; set; }

public byte IsDeleted { get; set; }

public void GetEmployeeDetails()

{

Console.WriteLine("Please enter employee name");

Name = Console.ReadLine();

Console.WriteLine("Please enter employee age");

int age;

while (!Int32.TryParse(Console.ReadLine(), out age))

{

Console.WriteLine("Invalid entry age. Please try again...");

}

Age = age;

}

public override string ToString()

{

return "Employee Id " + EmployeeId

+ "\nName " + Name

+ "\nAge " + Age

+ "\nIs Deleted? " + IsDeleted;

}

}

}